RECEIVED
OCT 23 1974

Postel

Network Working Group
Request for Comments: #644
NIC 30874

Bob Thomas, BBN-TENEX
July 22, 1974

On The Problem of Signature Authentication for Network Mail

This note describes the problem of signature authentication for network mail, presents a general approach to the problem and proposes a specific implementation of that approach.

1. The Problem

The problem we wish to consider is:

> How can the recipient of a network mail message be "certain" that the signature (e.g., the name in the "FROM" field) is authentic; that is, that the message is really from whom it claims to be?

We are interested in the problem of signature authenticity in the network context. For purposes of this note we shall assume a solution to the signature authentication problem for local mail (i.e., messages from one user to another within a single host). That is, we assume that for any host, either the host regards the problem as important and has a mechanism for guaranteeing signatures on local mail or that the host does not regard the problem as important and does not guarantee signature authentication. It should become clear how this assumption relates to our approach to the network signature problem.

We shall discuss our approach using the following simple model for network mail:

> To send net mail a user invokes a mail sending process (SP) on his local host (SH). The process SP acts on behalf of the user to deliver the message to an appropriate mailbox at the receiving host (RH). It does that by interacting with a receiving process (RP) that runs on host RH. RP accepts the message from SP and deposits it in the appropriate mailbox.

In the current implementation of network mail, the receiving process RP is typically an FTP server process. For the current TENEX implementation the mail sending process SP is either a process running SNDMSG or a "background" MAILER process which sends "queued" (previously posted but undelivered) mail.

2. An Approach

We seek a solution which will allow RP, the receiving process, to mark the signature on messages it receives as authenticated or not with respect to SH, the sending host. If RP can so mark incoming messages, a user reading his mail at RH would be able to see the signature on each message as authenticated or not with respect to the host of origin. The authenticity of the signature on a piece of mail is understood to be the responsibility of the originating host. The credibility a user gives a particular message which is marked as authentic can be based on the user's own estimate of the source host's user authentication and access control mechanisms.

The success of this approach depends upon two things:

a. Users develop estimates of the security of various host user authentication and access control mechanisms. We have seen that users who are concerned about data privacy and security are already doing this within the ARPANET.

b. The existence of a mechanism which RP, the receiving process, can use to distinguish mail authenticated with respect to the sending host from mail that has not been authenticated by the sending host. That is, a mechanism is required which will allow a properly authorized (by the sending host) mail sending process to identify itself as such to the mail receiving process. The receiving process can then mark mail from such an authenticated process as authentic. Non-authorized processes (e.g., a user process attempting to pose as an authorized mail sending process) may try to send mail to mailboxes at RH; in such a case the receiving process has the option of refusing to accept the messages or accepting them and marking them as unauthenticated.

3. Proposed Implementation of Approach

The use of passwords is one possible way to accomplish sending process authentication. Only an authorized sending process would know the password and thus be able to properly identify itself to a mail receiving process. We reject the password mechanism as operationally impractical for the following reasons:

a. Use of a password requires that the password be stored in the sending program or be accessible to it in some way thereby increasing the likelihood that the privacy of such a password will be compromised.

b. If a password is compromised, it must be changed at both sending and receiving hosts; a synchronization problem.

c. Truly secure mail would probably require passwords for each pair of hosts; this requires N*N passwords for an N host network.

As an alternative to the use of passwords as a means for process authentication, we propose that authentication be based on

the communication path itself between the sending and receiving processes. In the ARPANET, a communication path is uniquely identified by its two ends: the send host-socket pair and the receive host-socket pair. A process can accurately determine the host-socket pair at the remote end of a communication path. We propose that the receiving process consider the sending process to be a properly authorized (by the sending host) sender of mail only if the sending end of the communication path is (one of) the socket(s) reserved for transmission of authenticated mail. The mail sending socket(s) would be reserved by prior host agreement.

The responsibility of the sending host is to allow only authorized mail sending processes to access the mail sending socket(s). The responsibility of the mail receiving process is to mark as authenticated by the source host only mail received over the socket(s) reserved for mail. The responsibility of the user concerned about the authenticity of his mail is to understand that mail marked as authentic means that the sending host has determined the identity of the sender and that the signature on such mail is only as good or bad as the user authentication and access control procedures of the sending host.

the communication-path itself between the sending and receiving processes. In the ARPANET, a communication path is uniquely identified by its two ends: the send host-socket pair and the receive host-socket pair. A process can accurately determine the host-socket pair at the remote end of a communication path. We propose that the receiving process consider the sending process to be a properly authorized (by the sending host) sender of mail only if the sending end of the communication path is bound to the socket(s) reserved for transmission of authenticated mail. The mail sending socket(s) would be reserved by prior host agreement.

The responsibility of the sending host is to allow only authorized mail sending processes to access the mail sending socket(s). The responsibility of the mail receiving process is to mark as authenticated by the source host only mail received over the socket(s) reserved for mail. The responsibility of the user concerned about the authenticity of his mail is to realize that mail marked as authentic means that the sending host has ascertained the identity of the sender and that [illegible] is only as good as the user authentication and access control procedures of the sending host.

4. Additional Remarks

a. The use of sockets for process authentication is not a new concept within the ARPANET. By host agreement, the TELNET logger process responds to connections to socket #1, the FTP logger process to socket #3, etc. In fact, the privacy of net mail depends upon how well the receiving host controls access to the FTP logger socket; that is, the authenticity of the mail receiving process is based upon that fact that it is the process reached by ICP'ing to socket #3. This note proposes that the same mechanism be used to provide authentication of mail sending processes.

b. Planned TENEX experiment

A set of sockets has been assigned for mail transmission. They are (all numbers are decimal)

ICP "from" socket = 232
FTP user command sockets: receive, send = 234, 235
Default data transfer (user, send) socket = 237

We intend to modify the TENEX mail sending, receiving and reading software as suggested above. Mail sent by TENEX to remote hosts which is authentic (with respect to TENEX) will be sent by initiating the ICP to the remote FTP server from socket 232. Mail received from remote hosts will be marked as authentic only if the ICP to the TENEX FTP server was initiated from remote socket 232. The TENEX mail reading software will

indicate for each message whether or not the signature on the message was source authenticated.

c. Contention for the mail sending socket.

Depending upon the implementation of the sending host's NCP and its net mail sending software, it may be the case that several users concurrently sending network mail may be competing for the single ICP "from" socket. If socket contention turns out to be a serious problem in practice, a set of ICP "from" sockets could be reserved for authenticated network mail.

d. The local mail signature authentication problem is nearly independent of the network mail signature authentication problem as we have discussed it. For example, the following observations can be made:

1. The local users of a host which does not authenticate local mail probably should not expect the host to reliably deliver authenticated network mail to them. Because local mail is not authenticated, it is likely that a malicious local user could add to other users' mail boxes forged messages which are formatted identically to net mail and are marked as authentic in the way the host's mail receiving process marks mail.

2. A host that has strong user authentication procedures and authenticates local mail is not necessarily a reliable source of authenticated network mail. In order to be a reliable source, it must limit access to the net mail transmission socket(s) to authorized mail sending processes.

3. A host which does not support local authentic mail could be a reliable source of authentic net mail.

DRAFT

ARPANET expertise. The communications subnet and the hardware interfacing to the Hosts was subcontracted to a well-known hardware manufacturer with no known ARPANET expertise. (As an aside, but because it's so startling I can't forbear, the "system architect" for the target network is still another well-known hardware manufacturer (!), with, of course, no known ARPANET expertise.) To make a long, continuing story short, it is currently the case that the "real" ARPANET system whose hardware corresponds most closely to the machines being netted here (even though it is benchmarked at a rather lower "mips" (million intructions per second) rate than the target net's machinces) can transfer files at rates in excess of 28,000 bits per second (following the rather cumbersome full ARPANET FTP) from a small configuration development machine to a lightly loaded (but still running in excess of 20 users) service machine one Network "hop" away, while the new system achieves rates of no more than 4,000 bps even though only one process is being run on each machine -- also one "hop" away -- and the protocol for file transfer is nowhere near so general as in the ARPANET. Given a year or two, the situation can presumably be rectified, but at present it is fair -- if somewhat fanciful -- to say that if the Japanese were capable of only a like level of technology transfer they'd still be trying to make up their balance of trade with those cute little parasols on matchsticks.

Yet what has gone amiss here in Horror Story 2? I submit that the choice of subcontractors was based upon a misapprehension of the level of technological sophistication associated with the ARPANET, and that what was (is?) needed is a subcontract to a knowledgable ARPANET source (and I don't mean to the usual, profit-making place -- though I guess I trust them for the subnet), rather than to "outsiders". (I don't even mean to any particular place on the Net; maybe what's needed is to form a meta-place out of the whole Net. More on this, too, later.) The real point is that the model was essentially ignored by the putative model-followers, and -- demonstrably -- it shouldn't have been.

Case three should go a long way toward dispelling any impressions that might be building in the reader's mind that I'm some sort of hardcore ARPANET chauvinist. For even "insiders" have blown some. This is actually a dual case, for it involves two unsuccessful attempts to furnish terminal support mini-Hosts for the Net. In one case, the choice of machine was faulty; even with additional core memory field retrofitted, buffers cannot be furnished to support reasonable data rates without imposing considerable unnecessary Host overhead in the processing of too frequent Host-Host ALLocate commands. Nor is there enough room to furnish more than a rudimentary command language in the mini. Now these were knowledgable, reasonably well managed "insiders" -- but they were contractually not in a position to heed their own technical intuitions and the technical intuitions of most of their colleagues throughout the Network Working Group that they'd